Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 323–332

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

Case-based Reasoning for Knowledge Capitalization in Inventive Design Using Latent Semantic Analysis

Pei Zhang[a,*], Amira Essaid[c,d], Cecilia Zanni-Merk[b], Denis Cavallucci[a,c]

[a]CSIP research team @ ICube (UMR-CNRS 7357), 24 boulevard de la Victoire, 67084 Strasbourg Cedex, FRANCE
[b]INSA Rouen Normandie, LITIS, Normastic (FR CNRS 3638), Rouen, France
[c]INSA de Strasbourg, 24 boulevard de la Victoire, 67084 Strasbourg Cedex, France
[d]SDC Team, ICube (UMR CNRS 7357), Strasbourg, 67412, France

Abstract

Nowadays, innovation represents one of the most crucial factors driving the success of companies. The Theory of Inventive Problem Solving (also known as TRIZ) is a well-established method to facilitate systematic inventive design. Although, TRIZ allows solving inventive problems through a panoply of knowledge sources, it may make inventive problem solving a time-consuming, experience demanding process and lead to waste of resources of the companies. To avoid the use of these tools and to help new users in solving their inventive problems without completely mastering TRIZ, we propose in this paper an approach based on the use of the Case-based reasoning (CBR) in order to capitalize experience. CBR is a knowledge paradigm that solves a new problem by finding the old similar cases and reusing them. The retrieval is conducted in order to find the old similar cases, and the old solutions of the retrieved cases are adapted to solve the new problem. In this paper, a systematic three-level adaptation is proposed to reduce the effort required of the users in choosing the suitable solution to solve their problem. An example is used to illustrate in detail the proposed approach.

Peer-review under responsibility of KES International

Keywords: Case-based Reasoning; TRIZ; Inventive Design Theory; Latent Semantic Analysis; Experience Reuse

1. Introduction

Nowadays, innovation represents one of the most crucial factors driving the success of companies. In fact, companies need to innovate in order to satisfy customer requirements and to be distinguished in the global market. To face the evolution of the business environment, industries should create innovative products in a continuous way. For that purpose, more and more companies turn their attention to TRIZ (the theory of inventive problem solving)[1].

TRIZ is recognized as a suitable theory for solving inventive problems in different domains[2]. It differs from other techniques by its three fundamental principles: problems are raised because of the evolution laws and the resolution of the problems should respect the evolution laws; the resolution of a problem is eased by conveying it in terms of a

* Corresponding author. Tel.: +33-(0)3-8814-4848.
E-mail address: pei.zhang@insa-strasbourg.fr

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.004

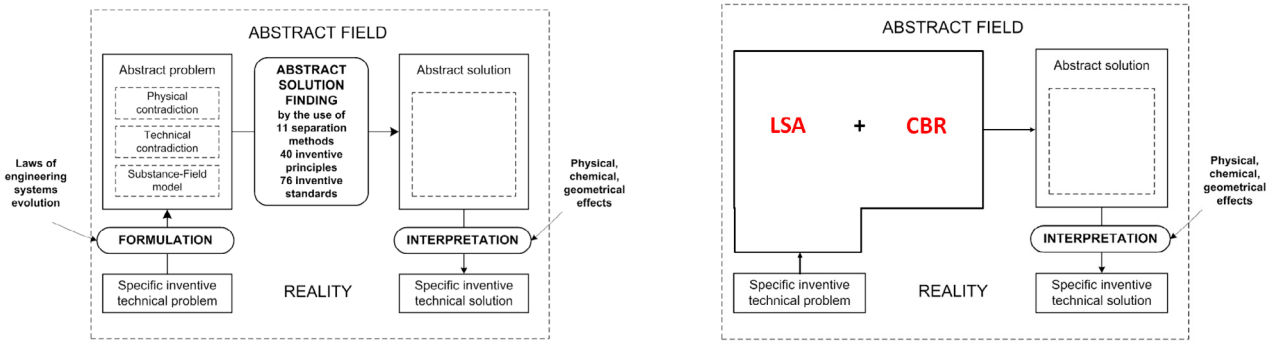

Fig. 1. (a) The approach of TRIZ solving inventive problem[3]; (b) The proposed approach.

contradiction; and a robust solution to a problem is considered to be a solution that minimizes the introduction and use of new resources. According to the TRIZ methodology, solving an inventive problem goes through three phases as illustrated in figure 1 (a):

1. The "formulation" phase, where the experts use different tools to formulate their specific inventive problem into an abstract problem in terms of a technical contradiction or other models.
2. The "abstract solution finding" phase, where, depending on the type of the problem of the former phase, knowledge bases are used to get one or more abstract solutions.
3. The "interpretation" phase, where the abstract solutions need to be interpreted by accessing to the scientific-engineering effects knowledge base in order to get one or more specific inventive technical solutions.

Although TRIZ offers to solve different types of inventive problems, but it is still difficult to use for novice users. In fact:

- TRIZ uses diverse knowledge sources at different abstraction levels. Depending on the abstraction level, it produces results also at different levels: at the highest abstraction level, the results are ideas of solutions (or concept solutions), at the lowest abstraction level (with the help of specific knowledge bases), it produces specific technical solutions.
- The wealth of knowledge available in TRIZ is necessary for solving a large variety of inventive problems but access to the needed specific knowledge might be difficult.
- TRIZ definitions are sometimes ambiguous and cannot be interpreted adequately. Using a recommendation proposed by TRIZ for solving a specific problem requires extensive knowledge of different engineering domains and is not currently supported by the methodology. As a consequence, the user is supposed to possess a high degree of expertise in engineering design.

In this paper, we study the feasibility of the use of case-based reasoning (CBR)[4] with latent semantic analysis (LSA)[5], in order to facilitate the TRIZ problem solving process (Figure 1 (b)), giving the users the possibility of obtaining an abstract solution directly from their concrete problems, avoiding then the error-prone process of abstracting the concrete problem to an abstract one.

The use of CBR permits the reuse of old solutions to solve new cases. The construction of semantic spaces with LSA[6] enables the automation of "analogy" between specific problems and abstract ones.

The remainder of this paper is as follows: section 2 gives a literature review of the current methods used in TRIZ to eliminate technical contradiction. Section 3 details each step of the proposed approach based on CBR. Section 4 studies the recycling bin problem to illustrate the proposed approach. Finally, section 5 sums with a conclusion and future perspectives.

2. Literature Review

A technical contradiction arises when it is required to improve some feature of the existing prototype but all solutions known within the domain do not produce the required result or their use would cause a negative effect. The impossibility to improve one parameter and to prevent another important parameter from deterioration is the main feature which separates inventive problems from problems that can be solved by a procedure of routine design.

Once the technical contradiction is identified at the concrete level, this model needs to be mapped into the abstract level, in terms of the 39 predefined *Generic Engineering Parameters* (or *GEPs*): "a *Generic Engineering Parameter* to be improved versus the other *Generic Engineering Parameter* which deteriorates"[7].

The problem can now be solved by the use of one of the 40 predefined *Inventive Principles* (*IP*)[8]. An *Inventive Principle* provides a guideline indicating in what way to solve a problem without causing negative effect. The principle itself does not give a solution to the problem; it recommends a method for eliminating a certain type of technical contradiction.

The choice of the good *Inventive Principle* can be done in a systematic way by accessing the more appropriate to solve the current problem through indices in a matrix, called the *Contradiction Matrix*[7]. Along the vertical axis of this matrix the *GEPs* have to be improved are specified. Along the horizontal axis the *GEPs* which deteriorate as a result of the improvement are specified. These *GEPs* can be looked up along the vertical and horizontal axes and the matrix suggests up to four principles that can be used to solve the contradiction. Selected principles are ordered according to their applicability. The principle that will most likely solve the problem is given first.

The TRIZ community describes the elimination of technical contradiction as an analogical process that leads to inventions[9]. The essential problems to use analogies successfully are of two types: one is the analogy reasoning between the specific problem and the abstract problem, the other is the analogy reasoning between the abstract problem and the abstract solution. To cope with the former problem, the work of[10] provided a way to systematically map design parameters (or *Specific Parameters* which are used to describe the specific problem) with the *Generic Engineering Parameters* in axiomatic design with TRIZ. While the author in[11] introduced the human factors issues in the *Generic Engineering Parameters* to ease the use of the *Contradiction Matrix*. To cope with the latter problem, ASIT[12] grouped 32 *Inventive Principles* into 5 thinking tools; the analysis conducted in[13] classified *Inventive Principles* into distinct principles and obscure principles. These approaches used the *Contradiction Matrix* for solving inventive problems; however, it is a time-consuming process.

In the last decades, the researchers started to point out the importance of combining Case-based reasoning (CBR) with TRIZ to reproduce human cognitive process for solving problems[14,15,16,17]. In order to improve the efficiency in the use of the the 40 *IPs*, the work in[15] used the 39 *Generic Engineering Parameters* (*GEPs*) of the *Contradiction Matrix* as an index for problems and the 40 *IPs* as solutions, but it lacks a proper problem analysis of the specific problem. Therefore, in[16], the authors proposed to add the function analysis for the selection of *GEPs*, while the work in[17] enhanced the approach of[16] by selecting retrieved cases with high reference value. In order to further address a thorough analysis of the problematic situation, the proposed work in[14] formulated the problem into a contradiction but raised the difficulty when associating the *Specific Parameters* with the *Generic Engineering Parameters*. To deal with it, the authors of[14] provided candidate *GEPs* for a certain *SP* by applying semantic techniques and left the user to choose an appropriate *GEP*. However, none of these works is able to avoid the abstract solution finding phase. Therefore, they fail to improve the efficiency of the problem solving process for new users. For that purpose, we propose to use CBR to help new users in solving inventive problems in an efficient way. The use of CBR goes through finding the similarity between the *SPs* of the old problems and the *SPs* of a new problem.

3. Case-based Reasoning for Inventive Problem Solving

The Case-based reasoning approach[4] is generally used to capture and induce experience by reproducing human problem solving process. The CBR approach comprises four activities:[18] Case representation; Case retrieval; Case adaptation; Case evaluation revision and learning. In the scope of this paper, we consider only the first three activities. The case evaluation revision and learning activity will be subject of future works.

Table 1. Features to describe a case.

	Input Features	Retrieval Features	Output Features
Problem Features	Element Action Parameter Value1 Value2	SP Degrading SP Improving	
Solution Features			Inventive Principle Inventive Principle No. Concept Solution

3.1. Case representation

Case-based reasoning adopts validated prior experience or "cases" as solutions to solve new problems[19]. A case contains different features (Table 1) grouped into three sets[20] according to their use:

- Input features are useful for text mining and for the complete description of a case.
- Retrieval features are useful for computing similarities between the old problems and the new one.
- Output features give information to the user.

As part of our approach, a case comprises a problem and its solution. The different features used to describe a problem and a solution are illustrated in Table 1. The second row of Table 1 details the features used to describe the problem. The retrieval features used for calculating similarities between the old problems and the new one are *SP degrading* and *SP improving*. The input features are those given by the user to describe a technical contradiction in a holistic way: *action parameter*, *element*, *value 1* (of *SP degrading*) and *value 2* (of *SP improving*). The third row of Table 1 describes the features of a solution. The output features of the solution include the *Inventive Principle* adopted to solve the specific technical contradiction as well as the *concept solution* designed by the experts based on the *Inventive Principle*.

3.2. Case retrieval

3.2.1. Semantic spaces construction

In our previous work, we proposed the use of LSA (Latent Semantic Analysis[1]) method to construct a geometrical semantic space and a physical semantic space with the aim of calculating the semantic similarity between *SPs*[6]. The two semantic spaces are constructed by finding the indexing words that are similar in meaning based on the mapping between *SPs* and *GEPs* and populating them by their synonyms. For example, ['area', 'length', 'volume', 'weight'] are some of the indexing concepts belonging to the geometrical semantic space and as example of indexing concepts of the physical semantic space, we may note ['vision', 'speed', 'efficiency', 'energy', 'illumination', 'intensity']. The constructed semantic spaces play an important role in finding the cases that are semantically similar.

3.2.2. Similarity calculations

The central issue of case retrieval is to find similarities between the new problem and the old problems. To calculate the similarity, first, we have to compare between the new *Specific Parameters* (SP_{new} improving and SP_{new} degrading) of the new problem and the old *Specific Parameters* (SP_{old} improving and SP_{old} degrading) of the problems stored in the case base. To do so, each *Specific Parameter* (the new one and the old one) is represented as a vector by adopting the Vector Space Model (VSM)[21]. Once the *Specific Parameters* have been transformed into vectors, usual metric space distances can be used such as the Euclidian distance that we choose to apply for its simplicity and suitability for our approach.

[1] also called Latent Semantic Indexing

The Euclidean distance is the square root of the sum of squared differences between the corresponding elements of the two vectors. In equation 1, we give the Euclidian distance between the two vectors representing respectively the ($SP_{new\ improving}$ and $SP_{old\ improving}$)

$$dist\left(\overrightarrow{SP}_{old\ improving}, \overrightarrow{SP}_{new\ improving}\right) = \sqrt{\sum_{i=1}^{n}\left(\overrightarrow{SP}_{old\ improving}[i] - \overrightarrow{SP}_{new\ improving}[i]\right)^2} \tag{1}$$

The Euclidean distance is also applied to calculate the similarity between vectors representing ($SP_{new\ degrading}$ and $SP_{old\ degrading}$).

As our aim is to resolve a new problem based on its similarity with the old problems, then a distance between a new problem and an old problem has to be calculated in order to depict the old problems similar to the new one. The similarity between a new problem and an old one represents a global similarity by taking weighted sums of local similarities (already calculated by the Euclidean distance) with the weights. As only the two features (*SP degrading* and *SP improving*) are taken into account for searching similarity between problems, we assign a weight of 0.5 to each local distance. In the equation 2, we give the global distance between a new problem (P_{new}) and an old problem (P_{old}) such as:

$$dist\,(P_{new}, P_{old}) = \frac{0.5 * dist_{SPimproving} + 0.5 * dist_{SPdegrading}}{0.5 + 0.5} \tag{2}$$

where $dist_{SPimproving}$ represents the local Euclidean distance between a SP improving of (P_{new}) and the SP improving of (P_{old}) whereas $dist_{SPdegrading}$ represents the local Euclidean distance between a SP degrading of (P_{new}) and the SP degrading of (P_{old}). The global distance of equation 2 is normalized between [0, 1] as presented in equation 3.

$$dist_{obtained}\,(P_{new}, P_{old}) = \frac{dist\,(P_{new}, P_{old}) - min\,(dist\,(P_{new}, P_{old}))}{max\,(dist\,(P_{new}, P_{old})) - min\,(dist\,(P_{new}, P_{old}))} \tag{3}$$

The global calculated similarities between the old problem of the old cases and the new problem focus only on similarity between *SPs* based on the letters composing the *SPs*. In order to enrich the results, it is interesting to consider also the cases that are similar in meaning. Based on the identified *SPs* similar in appearance to the *SPs* of the new problem, we search for each obtained *SP*, the similar terms in the constructed semantic spaces. For each found similar term, we run through the case base in order to identify the old cases that are similar in meaning to the new problem. Based on the obtained similar cases, the user chooses the cases that he wants to use for adaptation.

3.2.3. Case adaptation

In this section, we consider the reuse of old solutions by means of adaptation. Solving a new problem goes through either the direct reuse of old solutions stored in the case base or their adaptation to the new problem. In their work [18], the authors classified the adaptation into three types (null adaptation, transformational adaptation and generative adaptation), as illustrated in figure 2 based on [18]. In our approach, we adopt the three types as follows:

- Null adaptation: if the user finds that the obtained concept solution is useful and wishes to use it for the new problem, then no adaptation is applied and the new case will not be stored in the case base.
- Transformational adaptation: this type of adaptation can be further divided into two subtypes: substitutional and structural. The former consists in revising only the concept solution whereas the latter revises the solution at the *Inventive Principle* level by choosing a different *IP* and interpreting the concept solution accordingly.
- Generative adaptation: the user does not find any *Inventive Principle* useful to solve his problem so he reformulates his problem at a more detailed level using Su-Field model[2] and retrieve the solution in the Effects database[3].

In order to organize the transformational adaptation in a systematic way, we propose a three-level adaptation scheme to store the adapted solutions: the *IP groups level*, the *IP level* and the *concept solution level* as it is illustrated

[2] https://triz-journal.com/su-field-analysis/
[3] https://www.triz.co.uk/triz-effects-database

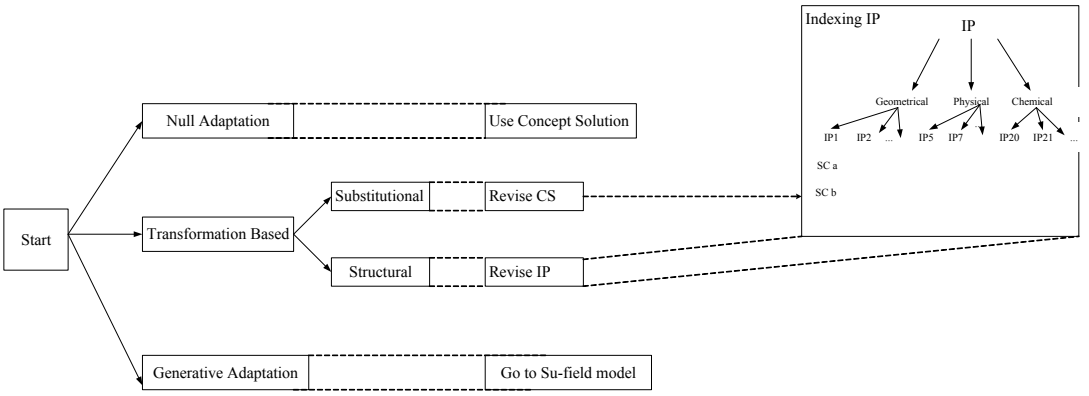

Fig. 2. Adaptation types in this approach.

in the rectangle at the right of figure 2 (based on[18]). We apply the LSA method[5] to group the 40 *IPs* into three groups (Geometrical, Physical and Chemical) and in each *IP group*, we store the *IPs* based on their common semantic meanings. The creation of the *IP groups level* aims to help the new user in choosing the suitable *IP* to solve his problem. Once we construct the *IP level*, we are able to store for each *IP*, the different *concept solutions* designed by the user. In this experiment, we used the corpus of *IP* as constructed in[22]. Following the steps of LSA described in[6], the obtained results are illustrated in figure 3 and table 2. Figure 3 presents the clusters of the different *IPs*, while table 2 presents the stemmed keywords of each topic associated to the *IPs*. In Figure 4, we illustrate the distribution of the cosine similarity between pairs of *IPs*. The x-axis and y-axis represent the number of the *IPs*, z-axis represents the cosine similarity between pairs of *IPs*. In order to cluster the *IPs* into the three groups (geometrical, physical and chemical), we used the bisection method to determine two thresholds. In addition to that, we assign three topics for the clusters of *IPs* in order to provide the user with the inventive direction for the solution:

- Thin, porous, three-dimensional, flexible part or structure;
- Thermal property, expansion, oxidizers, ionized or inert gas;
- Increase number, frequency or replace field.

4. Case study

In order to illustrate the proposed approach, we have developed a software prototype, with Java 1.8 as programming language, with a MySQL 5.7.14 database on a Windows environment. We used also myCBR [4] to build our CBR system.

The initial case base is composed of 47 old cases that has been solved by engineering students and experts in different domains. These domains include human necessities; performing operations, transporting; chemistry, metallurgy; textiles, paper; fixed constructions; mechanical engineering, lighting, heating, weapons, blasting; physics and electricity. To evaluate the feasibility of our proposal, we will show step by step how to solve a new specific problem:

> *Recycling bins are commonly seen in public spaces with the function of sorting glass waste and normal waste in order to better recycle them. A recycling bin is composed of two parts, one for containing glass waste and the other for containing normal waste. To keep the part of the bin related to the normal waste sanitary, the plastic garbage bags are useful to line the insides of this part. In order to reduce the plastic consumption, the body of the bin should be small to adapt it to smaller garbage bags. However, the body of the bin should be also big at the same time to satisfy the storage volume.*

[4] http://www.mycbr-project.net/

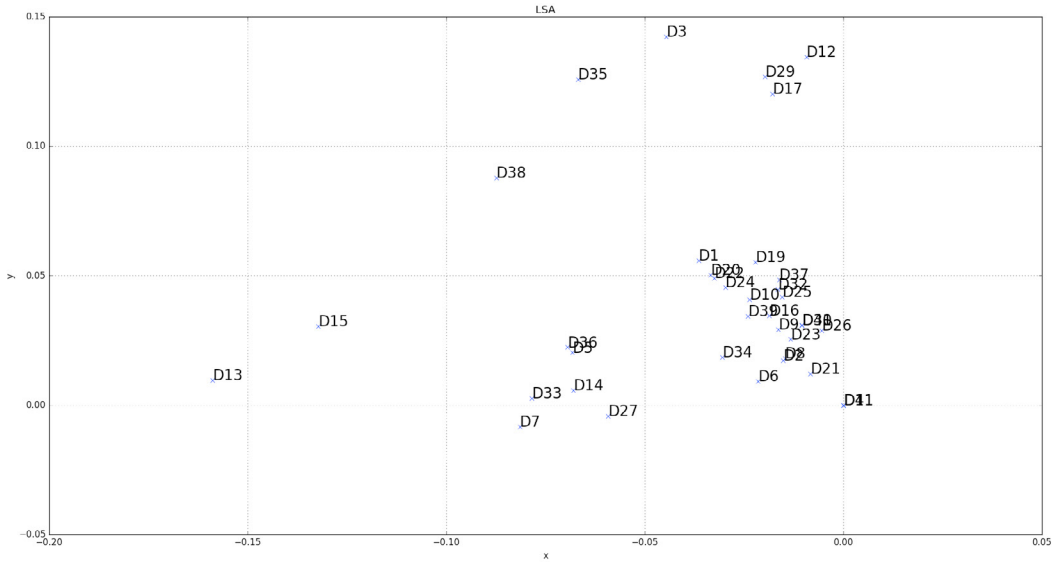

Fig. 3. LSA result of IP clustering in the two-dimensional space

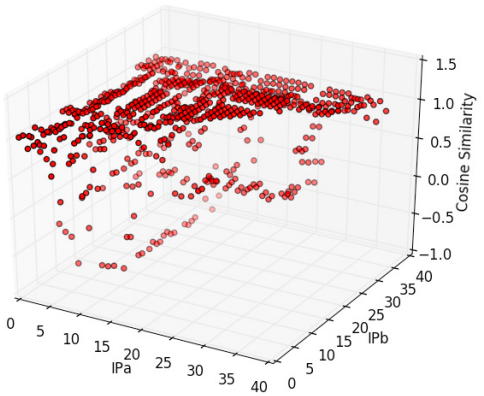

Fig. 4. Cosine similarity between each two IPs.

The body (*AP*) of the bin (*element*) has to be small (*value1*) to satisfy the *less plastic consumption* (*SP degrading*) and at the same time big (*value2*) to satisfy the *storage volume* (*SP improving*). As a consequence, we aim to use the proposed approach to resolve the specific problem which consists in increasing the *storage volume* (*SPDegrading*) without raising the *plastic consumption* (*SPImproving*). The resolution of this problem is handled in several steps:

- Case representation: the two new *Specific Parameters* of our "recycling bin" problem represents the retrieval features. They are used to retrieve similar cases.
- Case retrieval: Based on the proposed retrieval method detailed in 3.2.2, the system will first calculate the local similarity in appearance between the *storage volume* and all the *SPImproving* of the old cases as well as the local similarity in appearance between the *plastic consumption* and all the *SPDegrading* of the old cases. Then, the global similarity is calculated in order to obtain the cases that are similar in appearance with the specified

Table 2. IP topics.

Geometrical	Topic	Thin, porous, three-dimensional, flexible part or structure
	Documents	D1,D2,D3,D4,D6,D8,D9,D10,D11,D12,D13,D14,D15,D16,D17,D19,D23,D26,D30,D31,D32,D33,D34,D36,D40
	Terms	vari porou thin three-dimension shell film structur part weight flexibl forc action away chang color compens composit continu copi divid dynam elimin environ factor feedback forc prior redesign move period take
Chemical	Topic	Thermal property, expansion, oxidizers, ionized or innert gas
	Documents	D20,D21,D22,D24,D25,D29,D35,D37,D38,D39
	Terms	thermal air oxid ioniz function action expans transit phase properti inert degre harm materi strong
Physical	Topic	Increase number, frequency or replace field
	Documents	D5,D7,D18,D27,D28
	Terms	cheap combin electromagnet field fix frequenc increas interact life mechan motion movabl nest physic principl vibrat replac compon

problem. In figure 5, we illustrate the obtained old cases similar in appearance to our "recycling bin" problem. For each case, we display its identifier as well as the degree of its similarity to our problem. It is possible also to display the SP degrading and SP improving for an old similar case. For example, if we take the case identified as 45_A, we can see that the similarity distance between this case and our problem is equal to 0.54. In addition to that, the information related to the SPs of the old cases is presented. In fact, the case 45_A is described by the ease of storage as a *SPDegrading* and less crushing as a *SPImproving*. To enrich the results with cases similar in meaning, we take the SPs of already identified similar cases and we run through the semantic spaces in order to identify the terms that are similar in meaning to the SPs. As it is presented in figure 6, the similar words in meaning to the two SPs (the ease of storage, less crushing) found in the semantic spaces are: size, amount, quantity .... Based on these obtained words, we run through the case base in order to search for cases containing these words. As it can be seen in figure 6, the similar cases in meaning are case 13- the extension cord case and case 8 - the table case. At the end of this activity, we return to the users the found similar cases in appearance and in meaning to the new problem.

- Case adaptation: the user can choose one of the obtained cases to solve his new problem by adapting the old concept solution with the help of the three-level adaptation proposed in 3.2.3. For example in this case, the user may like to adapt the concept solution of the extension cord case. He can modify directly on the concept solution level based on his new problem, for example, he chooses to design a garbage bin with adjustable volume. If the user prefers to revise his solution on the *IP* level, he can choose a new *IP* like IP1-separation that is in the same group as the *15-Dynamics* and designs his new concept solution based on *IP1-separation*. The corresponding concept solution will be organized according to the IP it is generated from.
- Case evaluation revision and learning: This step is out of the scope of this paper, it will be subjected to future works.

5. Conclusion and future perspectives

This paper explores a Case-based reasoning approach in order to facilitate the TRIZ problem solving process. In the proposed approach, a new inventive problem is solved by retrieving the past similar problems in the case base according to their similarity in meaning. Then the new solution is constructed by adapting the retrieved solution. When the user needs to solve a concrete problem, he first inputs his problem into the system. Then, retrieval features are identified in order to be used for retrieving the most similar cases in meaning. Each case comprises a problem and its solution. At the adaptation stage, the retrieved concept solution is adapted by user manually according to his specific problem in order to obtain a concrete solution.

The contribution of the proposed approach can be summed into different points. First, the use of the constructed semantic spaces based on LSA in retrieval automates the "analogy" between the specific problem and the abstract problem, such that new users can benefit from using TRIZ methodology for solving inventive problems by expressing their specific problem in natural language. Second, the concept solution which has been validated by the past successful case is obtained as an output for the users to interpret it into the specific inventive solution for their specific

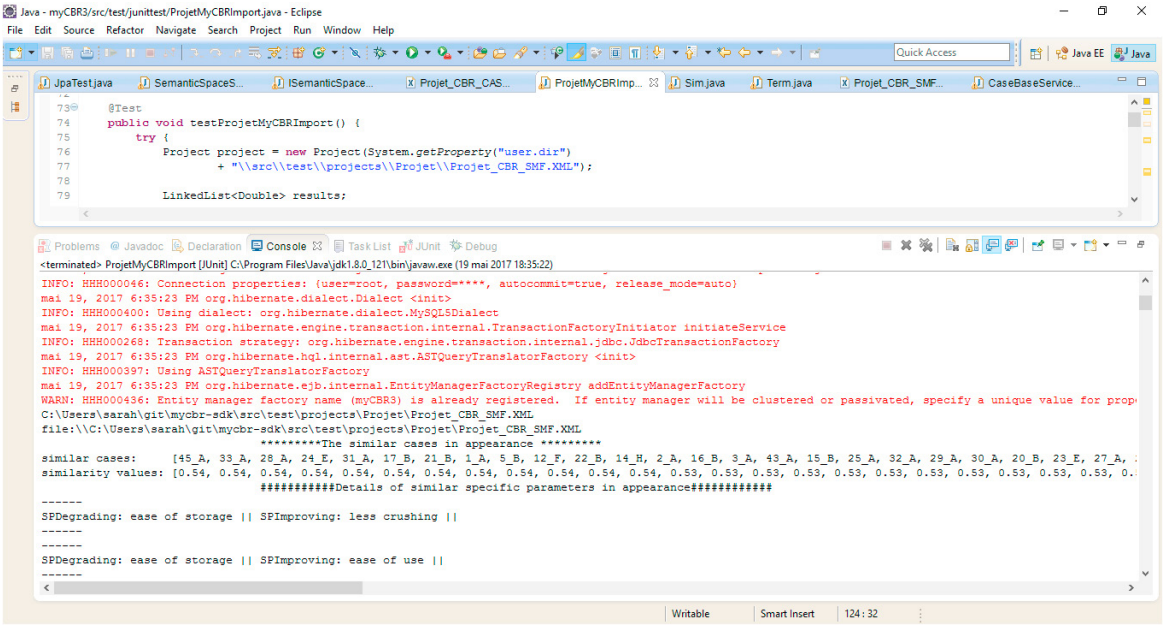

Fig. 5. Similar cases in appearance.

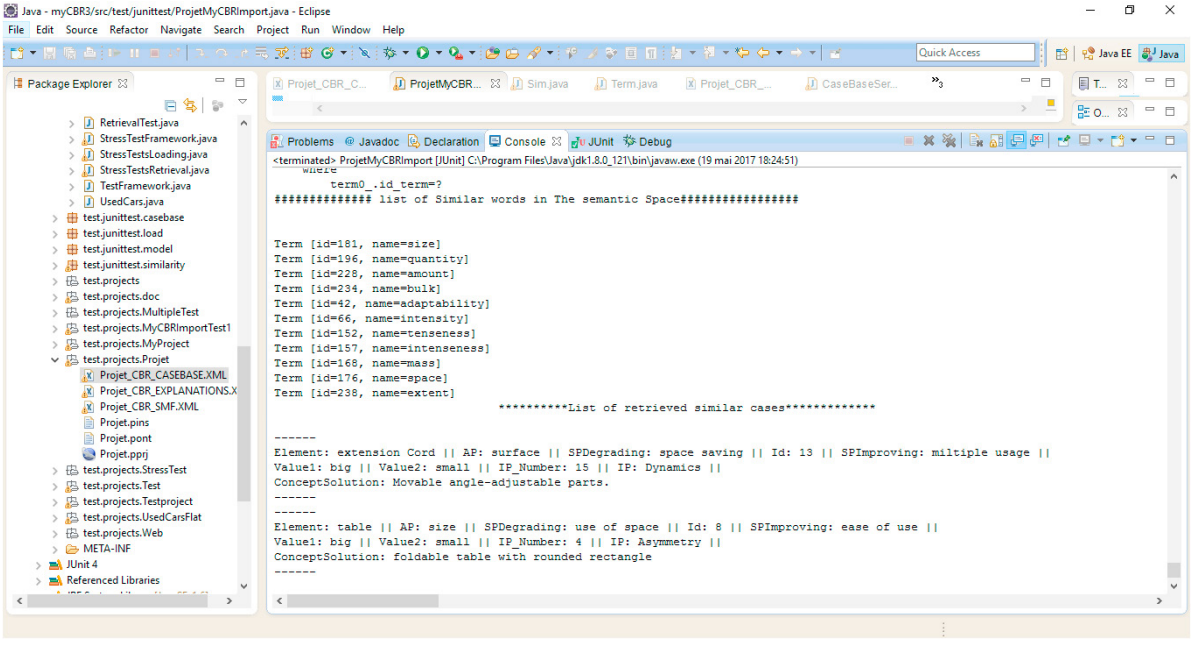

Fig. 6. Case retrieval result.

inventive problem. In addition to that, the non-use of some TRIZ tools such as the *Contradiction Matrix* improves considerably the efficiency of the inventive problem solving process without compromising the quality of the solution. Furthermore, the proposed three levels adaptation scheme allows more systematic experience reuse.

However, there are still two problems to be solved. One is that in this work we facilitate the use of the 40 *Inventive Principles* knowledge base but other knowledge sources that TRIZ provides should be also be included in the future work. For example, the use of the 76 *Inventive Standards* in order to solve problems that are formulated in terms of other models. The other is that the interpretation between the abstract solution and specific inventive solution is not

taken care of in this work. The interpretation of concept solution in order to generate a specific solution to use for the need of engineering application is generally implemented manually with the help of scientific-effects knowledge base. This scientific-effects knowledge base contains knowledge from different domains and should be systematically organized in order to obtain knowledge from different domains to improve the robustness of the specific solution.

References

1. Altshuller, G.S., Shulyak, L., Rodman, S.. *The innovation algorithm: TRIZ, systematic innovation and technical creativity*. Technical Innovation Center, Inc.; 1999.
2. Abramov, O.. Industry best practices and the role of TRIZ in developing new products. In: *International Research Conference TRIZFest*. 2013, p. 229–235.
3. Zanni-Merk, C., de Bertrand de Beuvron, F., Rousselot, F., Yan, W.. A formal ontology for a generalized inventive design methodology. *Applied Ontology* 2013;**8**(4):231–273.
4. Aamodt, A., Plaza, E.. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications* 1994;**7**(1):39–59.
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.. Indexing by latent semantic analysis. *Journal of the American society for information science* 1990;**41**(6):391.
6. Zhang, P., Zanni-Merk, C., Cavallucci, D.. Latent semantic indexing for capitalizing experience in inventive design. In: *International Conference on Sustainable Design and Manufacturing*; vol. 68. Springer; 2017, p. 37–47.
7. Domb, E.. The 39 features of Altshuller's contradiction matrix. The TRIZ Journal; 1998. [Online; https://triz-journal.com/; accessed 12-May-2017].
8. Mann, D., Domb, E.. 40 inventive principles with examples. The TRIZ Journal; 2013. [Available at http://www.triz-journal.com/archives/1997/07/b/].
9. Slocum, M.S.. Analogies are the way of breakthrough innovation. The TRIZ Journal; 2014. [Online; https://triz-journal.com/; accessed 12-May-2017].
10. Duflou, J.R., Dewulf, W.. On the complementarity of TRIZ and axiomatic design: from decoupling objective to contradiction identification. *Procedia Engineering* 2011;**9**:633–639.
11. Coelho, D.A.. Matching TRIZ engineering parameters to human factors issues in manufacturing. *Wseas Transactions on Business and Economics* 2009;**6**(11):547–556.
12. Toshio, T.. How people interact with objects using TRIZ and ASIT. The TRIZ Joural; 2003. [Available at http://www. triz-journal. com/archives/2003/08/d/].
13. Cong, H., Tong, L.H.. Grouping of TRIZ Inventive Principles to facilitate automatic patent classification. *Expert Systems with Applications* 2008;**34**(1):788–795.
14. Yan, W., Zanni-Merk, C., Rousselot, F., Cavallucci, D.. A method for facilitating inventive design based on semantic similarity and Case-based reasoning. *Procedia Engineering* 2015;**131**:194–203.
15. Cortes Robles, G., Negny, S., Le Lann, J.. Knowledge management and TRIZ: A model for knowledge capitalization and innovation. In: *World Conference: TRIZ Future*. 2004, .
16. Robles, G.C., Negny, S., Le Lann, J.M.. Case-based reasoning and TRIZ: A coupling for innovative conception in chemical engineering. *Chemical Engineering and Processing: Process Intensification* 2009;**48**(1):239–249.
17. Hu, Z., Zhao, Y., Chen, Y., Xiang, D.. An Improved TRIZ-CBR Model for Rapidly Innovative Design. In: *Advanced Materials Research*; vol. 308. Trans Tech Publ; 2011, p. 126–131.
18. Richter, M.M., Weber, R.. *Case-based reasoning: a textbook*. Springer Science & Business Media; 2013.
19. Houssin, R., Renaud, J., Coulibaly, A., Cavallucci, D., Rousselot, F.. TRIZ theory and Case based reasoning: synergies and oppositions. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 2015;**9**(3):177–183.
20. Richter, M.M.. Knowledge containers. Readings in Case-based reasoning. Morgan Kaufmann Publishers; 2003.
21. Salton, G., Wong, A., Yang, C.S.. A vector space model for automatic indexing. *Communications of the ACM* 1975;**18**(11):613–620.
22. Yan, W., Zanni-Merk, C., Cavallucci, D., Collet, P.. An ontology-based approach for inventive problem solving. *Engineering Applications of Artificial Intelligence* 2014;**27**:175–190.